

[illegible]

```

      AAAAAA  NN    NN    AAAAAA  LL      IIIIII  MM    MM  DDDDDDDD  MM    MM  PPPPPPPP
      AAAAAA  NN    NN    AAAAAA  LL      IIIIII  MM    MM  DDDDDDDD  MM    MM  PPPPPPPP
AA      AA  NN    NN    AA      AA  LL      II     MMMM  MMMM  DD      DD  MMMM  MMMM  PP      PP
AA      AA  NN    NN    AA      AA  LL      II     MMMM  MMMM  DD      DD  MMMM  MMMM  PP      PP
AA      AA  NNNN  NN    AA      AA  LL      II     MM  MM  DD      DD  MM  MM  PP      PP
AA      AA  NNNN  NN    AA      AA  LL      II     MM  MM  DD      DD  MM  MM  PP      PP
AA      AA  NN  NN  NN    AA      AA  LL      II     MM  MM  DD      DD  MM  MM  PPPPPPPP
AAAAAAAAAA  NN  NNNN  AAAAAAAAAA  LL      II     MM  MM  DD      DD  MM  MM  PPPPPPPP
AAAAAAAAAA  NN  NNNN  AAAAAAAAAA  LL      II     MM  MM  DD      DD  MM  MM  PP
AA      AA  NN    NN    AA      AA  LL      II     MM  MM  DD      DD  MM  MM  PP
AA      AA  NN    NN    AA      AA  LL      II     MM  MM  DD      DD  MM  MM  PP
AA      AA  NN    NN    AA      AA  LL      II     MM  MM  DD      DD  MM  MM  PP
AA      AA  NN    NN    AA      AA  LLLLLLLLLL  IIIIII  MM    MM  DDDDDDDD  MM    MM  PP
AA      AA  NN    NN    AA      AA  LLLLLLLLLL  IIIIII  MM    MM  DDDDDDDD  MM    MM  PP

```



```

LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SSSSSS
LL      II     SSSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS

```



```
0000 1      .TITLE ANALIMDMP
0000 2      .IDENT /V04-000/
0000 3
0000 4      *****
0000 5      *
0000 6      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8      *  ALL RIGHTS RESERVED.
0000 9      *
0000 10     *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11     *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12     *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13     *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14     *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15     *  TRANSFERRED.
0000 16     *
0000 17     *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18     *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19     *  CORPORATION.
0000 20     *
0000 21     *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22     *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23     *
0000 24     *
0000 25     *****
0000 26     :
0000 27     :
0000 28     :++
0000 29     : FACILITY:      IMAGE DUMP
0000 30     :
0000 31     : ABSTRACT:      Analyze an image dump and transfer control to debugger.
0000 32     :
0000 33     :
0000 34     : ENVIRONMENT:   User mode
0000 35     :
0000 36     : AUTHOR:        Wayne Cardoza
0000 37     :
0000 38     : CREATION DATE: 14-Feb-1983
0000 39     :
0000 40     : MODIFIED BY:
0000 41     :
0000 42     : V03-017 WMC0016      Wayne Cardoza  06-Aug-1984
0000 43     : Fix an improperly restored register.
0000 44     :
0000 45     : V03-016 WMC0015      Wayne Cardoza  09-Jul-1984
0000 46     : Save and restore CTLSGL_IMGHDRBF.
0000 47     :
0000 48     : V03-015 WMC0014      Wayne Cardoza  27-Jun-1984
0000 49     : Add control-Y handler to kill subprocess.
0000 50     :
0000 51     : V03-014 WMC0014      Wayne Cardoza  23-May-1984
0000 52     : Fix several minor bugs relating to error checks and reporting.
0000 53     :
0000 54     : V03-013 WMC0013      Wayne Cardoza  08-May-1984
0000 55     : Don't open dump file for write.
0000 56     :
0000 57     : V03-012 WMC0012      Wayne Cardoza  22-Mar-1984
```


0000 58 :
0000 59 :
0000 60 :
0000 61 :
0000 62 :
0000 63 :
0000 64 :
0000 65 :
0000 66 :
0000 67 :
0000 68 :
0000 69 :
0000 70 :
0000 71 :
0000 72 :
0000 73 :
0000 74 :
0000 75 :
0000 76 :
0000 77 :
0000 78 :
0000 79 :
0000 80 :
0000 81 :
0000 82 :
0000 83 :
0000 84 :
0000 85 :
0000 86 :
0000 87 :
0000 88 :
0000 89 :
0000 90 :
0000 91 :
0000 92 :

Don't let privilege be removed by image activation.

V03-011 WMC0011 Wayne Cardoza 29-Jan-1984
Fix defaults for /IMAGE.

V03-010 WMC0010 Wayne Cardoza 27-Dec-1983
Display the condition from the stack.

V03-009 WMC0009 Wayne Cardoza 13-Nov-1983
Don't let image activator remove privileges.

V03-008 WMC0008 Wayne Cardoza 26-Sep-1983
Vectors must also be reset after IMGACT.

V03-007 WMC0007 Wayne Cardoza 15-Sep-1983
Reset privileged library vectors when deleting P0.

V03-006 WMC0006 Wayne Cardoza 26-Aug-1983
Phony DEBUG frame had bad PUSH.

V03-005 WMC0005 Wayne Cardoza 14-Aug-1983
Fix priority of created subprocess.

V03-004 WMC0004 Wayne Cardoza 01-Jul-1983
SYSSIMGACT has been redesigned.

V03-003 WMC0003 Wayne Cardoza 25-May-1983
Fix a privilege problem.

V03-002 WMC0002 Wayne Cardoza 20-Apr-1983
Reset privileges before calling DEBUG.

V03-001 WMC0001 Wayne Cardoza 20-Apr-1983
Check dump version number for consistency.


```

00000000 94      .PSECT ANALIMDMP, LONG
          95      .DEFAULT DISPLACEMENT, WORD
          96      :
          97      : Data Structure Definitions
          98      :
          99      :
         100      $IMGDMPDEF
         101      $IMGMOVDEF
         102      $IODEF
         103      $DIBDEF
         104      $IACDEF
         105      $IHIDEF
         106      $IHDDEF
         107      $NAMDEF
         108      $PHDDEF
         109      $PRVDEF

```



```
0000 111 :++
0000 112 :
0000 113 Functional Description:
0000 114 This is the main routine for analyzing an image dump. It will display
0000 115 some useful data and then transfer control to the debugger.
0000 116 :
0000 117 Calling Sequence:
0000 118 standard
0000 119 :
0000 120 Input Parameters:
0000 121 standard image argument list
0000 122 :
0000 123 Implicit Inputs:
0000 124 the command line
0000 125 :
0000 126 Output Parameters:
0000 127 none
0000 128 :
0000 129 Implicit Outputs:
0000 130 none
0000 131 :
0000 132 Routine Value:
0000 133 none
0000 134 :
0000 135 Signals:
0000 136 none
0000 137 :
0000 138 Side Effects:
0000 139 many
0000 140 :
0000 141 --
0000 142 :
0000 143 :
0000 144 ANALIMDMP:
0000 145 .WORD 0
0000 146 :
17CB'CF 0C AC D0 0002 147 MOVL 12(AP),THIS_HDR ; Save this images pointers
17CF'CF 10 AC D0 0008 148 MOVL 16(AP),THIS_HDR+4 ; in case we don't load an image
0000 149 :
1908'CF 9F 000E 150 PUSHAB SYSS$INPUT ; Translate sys$input
193C'CF 9F 0012 151 PUSHAB INPUT_TRN
04E2 30 0016 152 BSBW GET_TRAN
1B1B 8F 1944'CF B1 0019 153 CMPW SYSS$INPUT_TRN,#<27+<27a8>> ; ESC-ESC means subprocess
08 12 0020 154 BNEQ 5$
03FA 30 0022 155 BSBW INIT_SUBP ; Do subprocess initialization
45 50 E9 0025 156 BLBC R0,10$
12 11 0028 157 BRB 7$
002A 158 :
13BC'CF 9F 002A 159 5$: PUSHAB DUMP_NAME ; Get the dump file name
18E2'CF 9F 002E 160 PUSHAB CLI_PARAMETER
00000000'GF 02 FB 0032 161 CALLS #2,G^CLIS$GET_VALUE
31 50 E9 0039 162 BLBC R0,10$
003C 163 :
003C 164 7$: $CREATE FAB = OUTFAB ; We will eventually need this
23 50 E9 0047 165 BLBC R0,10$
004A 166 $CONNECT RAB = OUTRAB
15 50 E9 0055 167 BLBC R0,10$
```


1358'CF	13BC'CF	90	0058	168 ;	
			0058	169 ;	
	01 50	E8	005F	170	
		04	006A	171	
			006D	172 10\$:	
	F1 50	E9	006E	173 20\$:	
			0079	174	
			007C	175 ;	
13AC'CF	01	D0	007C	176	
1394'CF	0200 8F	B0	0081	177	
1398'CF	1124'CF	9E	0088	178	
			008F	179	
	D0 50	E9	009A	180	
1B1B 8F	1944'CF	B1	009D	181	
	03	12	00A4	182	
	00A4	31	00A6	183	
52	1124'CF	9E	00A9	184 25\$:	
50	06 A2	3C	00AE	185	
	52 50	C0	00B2	186	
	50 62	9A	00B5	187	
1C64'CF	140C'CF	9E	00B8	188	
1C69'CF	50 04	81	00BF	189	
140C'CF	01 A2 50	28	00C5	190	
63	4558452E 8F	D0	00CC	191	
			00D3	192 ;	
	18EC'CF	9F	00D3	193	
00000000'GF	01	FB	00D7	194	
	20 50	E9	00DE	195	
	1614'CF	9F	00E1	196	
	18EC'CF	9F	00E5	197	
00000000'GF	02	FB	00E9	198	
	0E 50	E9	00F0	199	
1C68'CF	1614'CF	90	00F3	200	
1C60'CF	1618'CF	D0	00FA	201	
			0101	202 30\$:	
			0101	203	
	52 50	D0	010C	204	
0AF3'CF	1C8F'CF	9A	010F	205	
0AF7'CF	1C90'CF	D0	0116	206	
			011D	207	
	50 52	D0	0128	208	
	0F 50	E8	012B	209	
00000000'8F	50	D1	012E	210	
	79	12	0135	211	
	0AF3'CF	D4	0137	212	
	10	11	013B	213	
			013D	214 ;	
	18F9'CF	9F	013D	215 35\$:	
0000'CF	01	FB	0141	216	
	04 50	E9	0146	217	
	0AF3'CF	D4	0149	218	
			014D	219 ;	
13AC'CF	02	D0	014D	220 40\$:	
1394'CF	0200 8F	B0	0152	221	
1398'CF	140C'CF	9E	0159	222	
			0160	223	
	42 50	E9	016B	224	

MOVB	DUMP_NAME,DMP_FAB+FAB\$B_FNS	
\$OPEN	FAB = DMP_FAB	; Open the dump file
BLBS	RO,20\$	
RET		
\$CONNECT	RAB = DMP_RAB	
BLBC	RO,10\$	
MOVL	#1,DMP_RAB+RAB\$B_BKT	; Read the image header block
MOVW	#512,DMP_RAB+RAB\$W_USZ	; One block
MOVAB	IMGHDR,DMP_RAB+RAB\$B_UBF	
\$READ	RAB = DMP_RAB	
BLBC	RO,10\$	
CMPW	SY\$INPUT_TRN,#<27+<27a8>>	; ESC-ESC means subprocess
BNEQ	25\$	
BRW	40\$; Skip all the image name stuff
MOVAB	IMGHDR,R2	
MOVZWL	IND\$W_IMGIDOFF(R2),RO	; Get to image name
ADDL	RO,R2	
MOVZBL	IND\$T_IMGNAME(R2),RO	; Image name length
MOVAB	MISC,IMGFAB+FAB\$B_DNA	; Image name will be here
ADDB3	#4,RO,IMGFAB+FAB\$B_DNS	; Allow for .EXE in image name
MOVC3	RO,IND\$T_IMGNAME+1(R2),MISC	; Save image name for use as default
MOVL	#^A/.EXE7,(R3)	; Add default extension to end of MOVC3
PUSHAB	CLI_IMAGE	; See if image qualifier is there
CALLS	#1,G^CLIS\$PRESENT	
BLBC	RO,30\$	
PUSHAB	IMAGE_DESC	; Get image name
PUSHAB	CLI_IMAGE	
CALLS	#2,G^CLIS\$GET_VALUE	
BLBC	RO,30\$	
MOVB	IMAGE_DESC,IMGFAB+FAB\$B_FNS	
MOVL	IMAGE_DESC+4,IMGFAB+FAB\$B_FNA	
\$OPEN	FAB = IMGFAB	
MOVL	RO,R2	; Save status
MOVZBL	IMGNAME+NAM\$B_ESL,IMAGE	; Expanded file name length
MOVL	IMGNAME+NAM\$B_ESA,IMAGE+4	
\$CLOSE	FAB = IMGFAB	; We don't really need the file
MOVL	R2,RO	; Get back the status
BLBS	RO,35\$; The image file is there
CMPB	RO,#RMS\$_FNF	
BNEQ	50\$; A real error
CLRL	IMAGE	; Treat like /NOIMAGE
BRB	40\$	
PUSHAB	CLI_NOIMAGE	; Is noimage qualifier there
CALLS	#1,CLIS\$PRESENT	
BLBC	RO,40\$	
CLRL	IMAGE	; Indicate no image to be loaded
MOVL	#2,DMP_RAB+RAB\$B_BKT	; Read misc data block
MOVW	#512,DMP_RAB+RAB\$W_USZ	; One block
MOVAB	MISC,DMP_RAB+RAB\$B_UBF	
\$READ	RAB = DMP_RAB	
BLBC	RO,50\$	


```
03 1450'CF D1 016E 225 ;
09 13 016E 226 ;
50 00000000'8F D0 0173 227 ;
32 11 0175 228 ;
017C 229 ;
017E 230 ;
0845 30 017E 231 43$: BSBW BLD_MISC_VA ; Build table of misc VA's
0181 232 ;
1B1B 8F 1944'CF B1 0181 233 ;
05 13 0188 234 ;
0000'CF 00 FB 018A 235 ;
018F 236 ;
13AC'CF 03 D0 018F 237 45$: MOVL #3,DMP_RAB+RAB$BKT ; Read first map block
1394'CF 0200 8F B0 0194 238 ;
1398'CF 0F24'CF 9E 019B 239 ;
01A2 240 ;
01 50 E8 01AD 241 ;
04 01B0 242 50$: RET
01B1 243 60$: $DISCONNECT RAB = DMP_RAB
01BC 244 ;
01BF 245 ;
E3 50 E9 01CA 246 ;
01CD 247 ;
01CD 248 ;
01CD 249 ;
01CD 250 ;
0000000C'GF 144C'CF D1 01CD 251 ;
0D 1B 01D6 252 ;
1B1B 8F 1944'CF B1 01D8 253 ;
31 13 01DF 254 ;
0036 30 01E1 255 ;
04 01E4 256 ;
01E5 257 ;
1C18'CF 1D09'CF D0 01E5 258 70$: MOVL COND_MSG+4,OUTRAB+RAB$BKT ; First half of condition message
1C12'CF 1D05'CF B0 01EC 259 ;
01F3 260 ;
AF 50 E9 01FE 261 ;
5E 00001381'8F C2 0201 262 ;
6E 0561'CF 1381'8F 28 0208 263 ;
6E 17 0210 264 ;
0212 265 ;
0212 266 ;
0212 267 ;
50 00000000'8F D0 0212 268 80$: MOVL #SS$_VASFULL,R0
04 0219 269 ;
```

CMPL MISC+IMGDMP\$\$_VERSION,#IMGDMP\$\$_VERSION
BEQL 43\$; Versions of dump and program match
MOVL #SS\$_BADFILEVER,R0
BRB 50\$
BSBW BLD_MISC_VA ; Build table of misc VA's
CMPW SYSS\$INPUT_TRN,#<27+<27a8>> ; ESC-ESC means subprocess
BEQL 45\$; We already did the display
CALLS #0,DISPLAY_DUMP ; Display the dump data
MOVL #3,DMP_RAB+RAB\$BKT ; Read first map block
MOVW #512,DMP_RAB+RAB\$W_USZ ; One block
MOVAB MAP,DMP_RAB+RAB\$BKT
\$READ RAB = DMP_RAB
BLBS R0,60\$
RET
\$DISCONNECT RAB = DMP_RAB
BLBC R0,50\$
\$CLOSE FAB = DMP_FAB ; Close file before image activation
BLBC R0,50\$
; Decide if we can do the job in this process or if a subprocess is needed to
; make room for the saved stack.
CMPL MISC+IMGDMP\$\$_USRSTK,G^CTL\$\$_STACK+12
BLEQU 70\$; No problem
CMPW SYSS\$INPUT_TRN,#<27+<27a8>> ; ESC-ESC means subprocess
BEQL 80\$; Already a subprocess
BSBW CREATE_SUBP ; Go create a subprocess
RET
MOVL COND_MSG+4,OUTRAB+RAB\$BKT ; First half of condition message
MOVW COND_MSG,OUTRAB+RAB\$W_R5Z ; it is easier to do unrelocated
\$PUT RAB = OUTRAB
BLBC R0,50\$
\$MOVE_END-MOVE_BEG,SP
\$MOVE_END-MOVE_BEG,MOVE_BEG,(SP) ; Move the code
JMP (SP) ; Relocate execution
; No hope of analyzing this dump.
MOVL #SS\$_VASFULL,R0
RET


```
021A 271 :  
021A 272 : Create a subprocess to execute this image so we can fix the user stack in  
021A 273 : its old position.  
021A 274 :  
021A 275 CREATE_SUBP:  
021A 276 :  
021A 277 : Announce what we are doing  
021A 278 :  
1C18'CF 1CE8'CF D0 021A 279 MOVL CRE_SUB_MSG+4,OUTRAB+RAB$L_RBF  
1C12'CF 1CE4'CF B0 0221 280 MOVW CRE_SUB_MSG,OUTRAB+RAB$W_RSZ  
0228 281 $PUT RAB=OUTRAB  
1A 50 E9 0233 282 BLBC R0,10$  
0236 283 :  
0236 284 $CREMBX_S CHAN = INP_MBX,- ; Mailbox for new process SYSS$INPUT  
0236 285 MAXMSG = #256  
01 50 E8 024D 286 BLBS R0,20$  
05 0250 287 10$: RSB  
0251 288 20$: $CREMBX_S CHAN = TERM_MBX ; Termination mailbox for the created proces  
E9 50 E9 0264 289 BLBC R0,10$  
1908'CF 9F 0267 290 PUSHAB SYSS$INPUT ; Get recursive translation of SYSS$INPUT  
1984'CF 9F 026B 291 PUSHAB INPUT  
0289 30 026F 292 BSBW GET_TRAN  
5E 08 C0 0272 293 ADDL #8,SP  
D8 50 E9 0275 294 BLBC R0,10$  
1919'CF 9F 0278 295 PUSHAB SYSS$OUTPUT ; Get recursive translation of SYSS$OUTPUT  
19CC'CF 9F 027C 296 PUSHAB OUTPUT  
0278 30 0280 297 BSBW GET_TRAN  
5E 08 C0 0283 298 ADDL #8,SP  
C7 50 E9 0286 299 BLBC R0,10$  
192B'CF 9F 0289 300 PUSHAB SYSS$ERROR ; Get recursive translation of SYSS$ERROR  
1A14'CF 9F 028D 301 PUSHAB ERROR  
0267 30 0291 302 BSBW GET_TRAN  
5E 08 C0 0294 303 ADDL #8,SP  
B6 50 E9 0297 304 BLBC R0,10$  
1A5C'CF DD 029A 305 PUSHAB INP_MBX ; Get unit number of mailbox  
1A64'CF 9F 029E 306 PUSHAB INP_MBX_UNIT  
029D 30 02A2 307 BSBW MBX_UNIT  
5E 08 C0 02A5 308 ADDL #8,SP  
A5 50 E9 02A8 309 BLBC R0,10$  
1A60'CF DD 02AB 310 PUSHAB TERM_MBX ; Get unit number of mailbox  
1A68'CF 9F 02AF 311 PUSHAB TERM_MBX_UNIT  
028C 30 02B3 312 BSBW MBX_UNIT  
5E 08 C0 02B6 313 ADDL #8,SP  
94 50 E9 02B9 314 BLBC R0,10$  
02BC 315 $QIOW_S CHAN = INP_MBX,- ; Data for the new process  
02BC 316 FUNC = #10$ WRITEVBLK!IOSM_NOW,-  
02BC 317 P1 = @INPUT+4,-  
02BC 318 P2 = INPUT  
4C 50 E9 02DF 319 BLBC R0,30$  
02E2 320 $QIOW_S CHAN = INP_MBX,-  
02E2 321 FUNC = #10$ WRITEVBLK!IOSM_NOW,-  
02E2 322 P1 = @IMAGE+4,-  
02E2 323 P2 = IMAGE  
26 50 E9 0305 324 BLBC R0,30$  
0308 325 $QIOW_S CHAN = INP_MBX,-  
0308 326 FUNC = #10$ WRITEVBLK!IOSM_NOW,-  
0308 327 P1 = NAME_BUFFER,-
```



```
01 50      E8 0308 328
            05 032B 329
            032E 330 30$:
            032F 331 40$:
            032F 332
            032F 333
            032F 334
            18D6'CF 9F 0346 335
            18D2'CF 9F 034A 336
00000000'GF 02 FB 034E 337
            0355 338
            0355 339
            0366 340
            0366 341
            0366 342
            0387 343
            0387 344
            0387 345
            0387 346
            0387 347
            0387 348
            0387 349
            0387 350
            27 50 E9 0389 351
            038C 352
            038C 353
            038C 354
            038C 355
            05 50 E9 03DB 356
50 1ABB'CF D0 03DE 357
            50 DD 03E3 358 45$:
            7E D4 03E5 359
            18D6'CF 9F 03E7 360
00000000'GF 02 FB 03EB 361
            50 BA 03F2 362
            05 03F4 363 50$:
```

```
P2 = DUMP_NAME
R0,40$
BLBS
RSB
$FAO_S CTRSTR = INPFAO,- ; Get mailbox unit for SYSS$INPUT
        OUTLEN = INP_MBX_NAM,-
        OUTBUF = INP_MBX_NAM,-
        P1 = INP_MBX_UNIT
        OLD_CTRL
        CTRY_DISABLE
        CALLS #2,G^LIB$DISABLE_CTRL ; Disable DCL use of control-Y
        $ASSIGN_S DEVNAM = SYSS$INPUT,- ; Get channel for control-Y
        CHAN = INP_CHAN
        $QIO_S CHAN = INP_CHAN,- ; Request AST on control-Y
        FUNC = #IOS$SETMODE!IOSM_CTRL_YAST,-
        P1 = CNTRY_AST
        $CREPRC_S IMAGE = ANAC_IMG,- ; Create the process
        INPUT = INP_MBX_NAM,-
        OUTPUT = OUTPUT,-
        ERROR = ERROR,-
        MBXUNT = TERM_MBX_UNIT,-
        BASPRI = #4,-
        PRVADR = L^CTL$GQ_PROCPRIV,-
        PIDADR = SUBP_PID
        BLBC R0,45$
        $QIOW_S CHAN = TERM_MBX,-
        FUNC = #IOS$READVBLK,-
        P1 = TERM_MSG,-
        P2 = #10
        BLBC R0,45$
        MOVL TERM_MSG+4,R0 ; Get process exit status
        PUSH R0 ; Save exit status
        CLRL -(SP)
        PUSHAB OLD_CTRL ; Original control-Y status
        CALLS #2,G^LIB$ENABLE_CTRL ; Restore DCL use of control-Y
        POPR R0
        RSB
```



```
03F5 365 ;  
03F5 366 ; AST routine for control-Y when a subprocess is active  
03F5 367 ;  
03F5 368 CNTRLY_AST:  
0000 03F5 369 .WORD 0  
03F7 370 $DELPID _S PIDADR = SUBP_PID ; Get rid of the subprocess  
0404 371 CLRL -(SP)  
0406 372 PUSHAB OLD_CTRL ; Original control-Y status  
040A 373 CALLS #2, G^LIB$ENABLE_CTRL ; Restore DCL use of control-Y  
0411 374 $EXIT _S CODE = #SS$_NORMAL ; Exit this image  
04 041E 375 RET
```



```
041F 377 :  
041F 378 : Initialization routines for running in subprocess  
041F 379 :  
041F 380 INIT_SUBP:  
193C'CF 02 C2 041F 381 SUBL2 #2,INPUT_TRN ; Get rid of the ESC-ESC  
1940'CF 02 C0 0424 382 ADDL #2,INPUT_TRN+4  
0429 383 $ASSIGN_S DEVNAM = INPUT_TRN,-  
0429 384 CHAN = INP_MBX  
01 50 E8 043A 385 BLBS R0,20$  
05 043D 386 10$: PSB  
043E 387 20$: $QIOW_S CHAN = INP_MBX,-  
043E 388 FUNC = #IOS_READVBLK,-  
043E 389 IOSB = IOSB,-  
043E 390 P1 = @REAL_INPUT+4,-  
043E 391 P2 = REAL_INPUT  
50 D9 50 E9 0461 392 BLBC R0,10$  
184D'CF 3C 0464 393 MOVZWL IOSB,R0  
D1 50 E9 0469 394 BLBC R0,10$  
1B55'CF 1B4F'CF B0 046C 395 MOVW IOSB+2,REAL_INPUT ; Length  
0473 396 $CRELOG_S LOGNAM = SYSSINPUT,- ; Make SYSSINPUT correct  
0473 397 EQLNAM = REAL_INPUT,-  
0473 398 TBLFLG = #2  
B4 50 E9 0486 399 BLBC R0,10$  
0489 400 $QIOW_S CHAN = INP_MBX,-  
0489 401 FUNC = #IOS_READVBLK,-  
0489 402 IOSB = IOSB,-  
0489 403 P1 = @IMAGE_DESC+4,-  
0489 404 P2 = IMAGE_DESC  
50 4B 50 E9 04AC 405 BLBC R0,30$  
184D'CF 3C 04AF 406 MOVZWL IOSB,R0  
43 50 E9 04B4 407 BLBC R0,30$  
0AF3'CF 1B4F'CF B0 04B7 408 MOVW IOSB+2,IMAGE ; Length  
0AF7'CF 1618'CF D0 04BE 409 MOVL IMAGE_DESC+4,IMAGE+4  
04C5 410 $QIOW_S CHAN = INP_MBX,-  
04C5 411 FUNC = #IOS_READVBLK,-  
04C5 412 IOSB = IOSB,-  
04C5 413 P1 = @DUMP_NAME+4,-  
04C5 414 P2 = DUMP_NAME  
50 0F 50 E9 04E8 415 BLBC R0,30$  
184D'CF 3C 04EB 416 MOVZWL IOSB,R0  
07 50 E9 04F0 417 BLBC R0,30$  
13BC'CF 1B4F'CF B0 04F3 418 MOVW IOSB+2,DUMP_NAME ; Length  
05 04FA 419 30$: RSB
```



```
04FB 421 :
04FB 422 : Misc routines for subprocess creation
04FB 423 :
04FB 424 :
04FB 425 : Recursively translate a logical name
04FB 426 : Inputs
04FB 427 : address of descriptor of output buffer
04FB 428 : address of descriptor of input name
04FB 429 :
04FB 430 GET_TRAN:
04FB 431      MOVL 4(SP),R2      ; Output descriptor
04FF 432      MOVQ (R2),LOG_OUT ; Output descriptor
0504 433      MOVQ @8(SP),LOG_IN ; Name to be translated
050A 434 10$: STRNLOG_S LOGNAM = LOG_IN,-
050A 435      RSLLEN = (R2),-
050A 436      RSLBUF = LOG_OUT
0521 437      BLBC R0,20$
0524 438      CMPL R0,#SS$_NOTRAN
052B 439      BEQL 20$      ; All done
052D 440      MOVQ (R2),LOG_IN ; Result of the last try
0532 441      BRB 10$
0534 442 20$: CMPW @4(R2),#27 ; ESC-0 means PPF
0538 443      BNEQ 30$
053A 444      SUBL #4,(R2) ; Get rid of PPF header
053D 445      ADDL #4,4(R2)
0541 446 30$: RSB
0542 447 :
0542 448 :
0542 449 : Get a mailbox unit number
0542 450 : Inputs
0542 451 : address to return unit number
0542 452 : channel number
0542 453 :
0542 454 MBX_UNIT:
0542 455      MOVL 8(SP),R1 ; Channel
0546 456      $GETCHN_S CHAN = R1,-
0546 457      PRIBUF = MBXCHAR
055A 458      MOVZWL MBXCHARBUF+DIB$_UNIT,@4(SP)
0560 459      RSB
0561 460
```

52 04 AE D0 04FB 421
1AC1'CF 62 7D 04FF 422
1AC9'CF 08 BE 7D 0504 423
00000000'8F 10 50 E9 0521 437
07 50 D1 0524 438
1AC9'CF 07 13 052B 439
62 7D 052D 440
1B 04 B2 11 0532 441
07 12 0534 442
62 04 C2 0538 443
04 A2 04 C0 053A 444
05 053D 445
0541 446
0542 447
0542 448
0542 449
0542 450
0542 451
0542 452
0542 453
51 08 AE D0 0542 454
04 BE 1AE5'CF 3C 0546 456
05 0546 457
055A 458
0560 459
0561 460


```
0561 462 :
0561 463 : The relocateable portion of the code begins here
0561 464 :
0561 465 MOVE_BEG:
57 5E 00000561'8F C3 0561 466 SUBL3 #MOVE_BEG,SP,R7 ; Relocation constant
01 50 E8 0569 467 $CMKRNLS_ROUTIN = DELETE ; Go delete P0 before image activation
0A98'CF 0000000C'9F D0 0576 468 BLBS RO,10$
0A9C'CF 00000000'9F D0 0579 469 RET
OAF3'CF 2B D5 057A 470 10$: MOVL @#CTLSAL_STACK+12,STACK INI ; Save current stack base
OAF7'CF 57 C0 0583 471 MOVL @#CTLSGL_IMGHDRBF,IMGHDRBF INI ; Save pointer for analimdp
OAC8'CF 57 C0 058C 472 TSTL IMAGE ; Should we load an image
0A 50 E8 0590 473 BEQL 40$ ; No
00000000'8F 50 D1 0592 474 ;
08 13 0592 475 ADDL R7,IMAGE+4 ; Relocate image name
OAF7'CF 57 C0 0597 476 ADDL R7,IMG_DEFAULT+4
OAC8'CF 57 C0 059C 477 $CMEXEC_S_ROUTIN = IMGACT ; IMGACT it in EXEC mode so we are
0A 50 E8 05A9 478 BLBS RO,30$ ; able to restore privileges
00000000'8F 50 D1 05AC 479 CMPL RO,#RMS$_FNF
08 13 05B3 480 BEQL 40$ ; Treat FNF like /NOIMAGE
04 05B5 481 RET
05B6 482 30$: $IMGFIX_S ; Address fixups
05BD 483 :
01 50 E8 05BD 484 40$: $CMKRNLS_ROUTIN = RESET_VEC ; Reset any privileged library vectors
04 05CA 485 BLBS RO,45$
05CD 486 RET
0DOF'CF 57 C0 05CE 487 45$: ADDL R7,DEBUG+4 ; Relocate DEBUG name
OADF'CF 57 C0 05D3 488 ADDL R7,DBG_DEFAULT+4
OD03'CF 171B'CF D0 05D8 489 MOVL DEBUG VA,DBG_RANGE ; First unallocated P0
05DF 490 $IMGACT_S NAME = DEBUG,- ; Merge in DEBUG
05DF 491 DFLNAM = DBG_DEFAULT,-
05DF 492 IMGCTL = #IACSM_MERGE,-
05DF 493 INADR = DBG_RANGE,-
05DF 494 RETADR = DBG_RETADR,-
05DF 495 HDRBUF = DBG_HDRBUF
32 50 E9 0600 496 BLBC RO,50$
0603 497 $IMGFIX_S
060A 498 ;
13B0'CF 57 C0 060A 499 ADDL R7,DMP_RAB+RAB$_FAB ; Relocate file name stuff
1350'CF 57 C0 060F 500 ADDL R7,DMP_FAB+FAB$_FNA
1354'CF 57 C0 0614 501 ADDL R7,DMP_FAB+FAB$_DNA
0E 50 E9 0619 502 $OPEN FAB = DMP_FAB ; Open the dump file again
0624 503 BLBC RO,50$
0627 504 $CONNECT RAB = DMP_RAB
01 50 E8 0632 505 BLBS RO,60$
03C2 30 0635 506 50$: RET
0636 507 60$: BSBW RESTORE_MISC_VA ; Restore the misc VA
0639 508 :
0639 509 : Do original process address space
0639 510 :
13AC'CF 04 D0 0639 511 MOVL #4,DMP_RAB+RAB$_BKT ; First data block
55 OF24'CF 9E 063E 512 MOVAB MAP,R5 ; Map pointer
52 85 D0 0643 513 70$: MOVL (R5)+,R2 ; Page count
03 18 0646 514 BGEQ 80$
0283 30 0648 515 BSBW NEXT_MAP ; Nothing left in this map block
03 12 064B 516 80$: BNEQ 90$
016B 31 064D 517 BRW GET_DEBUG ; All done - go to DEBUG
03 65 1E E1 0650 518 90$: BBC #30,(R5),95$
```



```
0084 31 0654 519
1398'CF 85 D0 0657 520 95$: BRW GET P1 ; Go restore P1
53 52 00000200 8F C5 065C 521 100$: MOVL (R5)+,DMP_RAB+RAB$$_UBF ; Starting VA
0000FE00 8F 53 D1 0664 522 CMPL R3,#<127*512> ; Byte count
07 1B 066B 523 BLEQU 101$ ; Is it greater than maximum
53 0000FE00 8F D0 066D 524 MOVL #<127*512>,R3 ; No
160C'CF 1398'CF D0 0674 525 101$: MOVL DMP_RAB+RAB$$_UBF,CREATE_PAGE ; Set up to create the page
1610'CF 1398'CF 53 C1 067B 526 ADDL3 R3,DMP_RAB+RAB$$_UBF,CREATE_PAGE+4
1610'CF 1610'CF D7 0683 527 DECL CREATE_PAGE+4 ; Top of range
0687 528 $CRETVA_S INADR = CREATE_PAGE -
0687 529 RETADR = CREATE_PAGE
15 50 E8 0698 530 BLBS R0,110$ ; Were there any problems
53 1610'CF 160C'CF C3 069B 531 ; Not owner of page
09 12 06A3 532 SUBL3 CREATE_PAGE,CREATE_PAGE+4,R3 ; Created byte count - 1
53 00000200 8F D0 06A5 533 BNEQ 105$ ; Any pages created?
15 11 06AC 534 MOVL #512,R3 ; Skip one page
53 1394'CF 53 D6 06AE 535 BRB 115$ ; Avoid the read
105$: INCL R3 ; Make it byte count
110$: MOVW R3,DMP_RAB+RAB$$_USZ
$READ RAB = DMP_RAB ; Read the dump page
17 50 E9 06C0 538 BLBC R0,120$
53 1398'CF 53 C0 06C3 539 ADDL R3,DMP_RAB+RAB$$_UBF
53 53 F7 8F 78 06C8 540 115$: ASHL #-9,R3,R3 ; Page count
13AC'CF 53 C0 06CD 541 ADDL R3,DMP_RAB+RAB$$_BKT
52 53 C2 06D2 542 SUBL R3,R2 ; Remaining page count
85 14 06D5 543 BGTR 100$
FF69 31 06D7 544 BRW 70$
04 06DA 545 RET
06DB 546 120$:
06DB 547 GET_P1:
50 0B03'CF 9E 06DB 548 MOVAB IMG_HDRBUF,R0 ; Normalize image header
0B03'CF 50 C2 06E0 550 SUBL R0,IMG_HDRBUF
0B07'CF 50 C2 06E5 551 SUBL R0,IMG_HDRBUF+4
13B0'CF 57 C2 06EA 552 SUBL R7,DMP_RAB+RAB$$_FAB ; and the RAB
55 04 C2 06EF 553 SUBL #4,R5 ; Reset map pointer
06F2 554
06F2 555 ; Space is created and the code is relocated to the top of P0
06F2 556 ; A P0 stack (one page) is also created
06F2 557
06F2 558 $EXPREG_S PAGCNT = #<<MOVE_END-MOVE_BEG>+511/512>+1,-
06F2 559 RETADR = NEW_PO
1404'DF FE51 CF 1381'8F 28 0707 560 PUSHL R5
55 8E D0 0709 561 MOVCL3 #MOVE_END-MOVE_BEG,MOVE_BEG,@NEW_PO
50 0F24'CF 9E 0713 562 MOVL (SP)+,R5
55 50 C2 0716 563 MOVAB MAP,R0
50 1404'CF D0 071B 564 SUBL R0,R5 ; Normalize map pointer
000001C8'E0 17 071E 565 MOVL NEW_PO,R0 ; Address code was moved to
57 50 00000561'8F C3 0723 566 JMP <10$-MOVE_BEG>(R0) ; Relocate execution
1380'CF 57 C0 0729 567 10$: SUBL3 #MOVE_BEG,R0,R7 ; Relocation constant
50 0F24'CF 9E 0729 568 ADDL R7,DMP_RAB+RAB$$_FAB
55 50 C0 0731 569 MOVAB MAP,R0
50 0B03'CF 9E 0736 570 ADDL R0,R5 ; Relocate map pointer
0B03'CF 50 C0 073B 571 MOVAB IMG_HDRBUF,R0 ; and image header
0B07'CF 50 C0 073E 572 ADDL R0,IMG_HDRBUF
5E 1408'CF D0 0743 573 ADDL R0,IMG_HDRBUF+4
0748 574 ADDL R0,IMG_HDRBUF+4
074D 575 MOVL NEW_PO+4,SP ; Get stack out of the way of P1
```



```
0760'CF 00 FB 0752 576 CALLS #0,15$ ; Create top frame for new stack
0757 577 $EXIT_S CODE = R0 ; Get out with correct status
0760 578
0000 0760 579 15$: .WORD 0
6D D4 0762 580 CLRL (FP) ; Terminate frames
0764 581
52 85 D0 0764 582 20$: MOVL (R5)+,R2 ; Page count
03 18 0767 583 BGEQ 30$
0162 30 0769 584 BSBW NEXT_MAP ; We have finished this map block
4D 13 076C 585 30$: BEQL GET_DEBUG ; All done - go start DEBUG
1398'CF 85 D0 076E 586 30$: MOVL (R5)+,DMP_RAB+RAB$$_UBF ; Starting VA
160C'CF 1398'CF D0 0773 587 40$: MOVL DMP_RAB+RAB$$_UBF,CREATE_PAGE ; Set up to create the page
1610'CF 1398'CF D0 077A 588 40$: MOVL DMP_RAB+RAB$$_UBF,CREATE_PAGE+4
0781 589 $CRETVA_S INADR = CREATE_PAGE
0790 590 BLBC R0,50$ ; Not owner of page
1394'CF 0200 8F B0 0793 591 MOVW #512,DMP_RAB+RAB$$_USZ ; One block
079A 592 $READ RAB = DMP_RAB
12 50 E9 07A5 593 BLBC R0,50$
13AC'CF D6 07A8 594 INCL DMP_RAB+RAB$$_BKT ; Next file block
1398'CF 00000200 8F C0 07AC 595 ADDL #512,DMP_RAB+RAB$$_UBF ; Next page
BB 52 F5 07B5 596 SOBGTR R2,40$
AA 11 07B8 597 BRB 20$ ; Go do next run of pages
07BA 598
04 07BA 599 50$: RET
07BB 600
07BB 601 GET_DEBUG:
07BB 602 $CMKRNL_S ROUTIN = FIX_STACK ; Put stack limits back
07C8 603 $CMKRNL_S ROUTIN = FIX_IMGHDRBF ; Restore CTL$GL_IMGHDRBF
07D5 604 $CMKRNL_S ROUTIN = RESET_PRIV ; Eliminate the image privileges
07E2 605 TSTL IMAGE ; Was an image loaded
07E6 606 BNEQ 10$
07E8 607 MOVL THIS_HDR,DBG_ARG+12 ; Let DEBUG try to look at this image
07EF 608 MOVL THIS_HDR+4,DBG_ARG+16 ; just to keep it happy
07F6 609 BRB 20$
07F8 610 10$: MOVL IMG_HDRBUF,DBG_ARG+12 ; Arguments to start DEBUG
07FF 611 MOVL IMG_HDRBUF+4,DBG_ARG+16
0806 612 20$: MOVL DBG_RETADR,R0
080B 613 MOVAB @8(R0)[R0],DEBUG_BEG ; Find debug transfer address
0812 614
0812 615 ; Display the original cause of the dump
0812 616
0812 617 MOVL MISC+IMGDMP$$_AP,R2
0817 618 BISL #XFA16,@4(R2) ; Add flags to display all message fields
081F 619 $PUTMSG_S MSGVEC = @4(R2) ; Ignore errors
082F 620 BICL #XFA16,@4(R2) ; Clear the flags
0837 621
0837 622 MOVL MISC+IMGDMP$$_SP,SP ; Restore registers
083C 623 MOVL MISC+IMGDMP$$_AP,AP
0841 624 MOVQ MISC+IMGDMP$$_R0,R0
0846 625 MOVQ MISC+IMGDMP$$_R2,R2
084B 626 MOVQ MISC+IMGDMP$$_R4,R4
0850 627 MOVQ MISC+IMGDMP$$_R6,R6
0855 628 MOVQ MISC+IMGDMP$$_R8,R8
085A 629 MOVQ MISC+IMGDMP$$_R10,R10
085F 630
085F 631 ; Build a phony SS$_DEBUG frame
085F 632 ;
```



```
50 04 AC DD 085F 633 MOVL 4(AP),R0 ; Real signal array
51 60 DD 0863 634 MOVL (R0),R1
6041 DD 0866 635 PUSHL (R0)[R1] ; PC, PSL
FC A041 DD 0869 636 PUSHL -4(R0)[R1]
00000000'8F DD 086D 637 PUSHL #SS$_DEBUG
03 DD 0873 638 PUSHL #3
51 5E DD 0875 639 MOVL SP,R1
1410'CF DD 0878 640 PUSHL MISC+IMGDMP$R1
140C'CF DD 087C 641 PUSHL MISC+IMGDMP$R0
7E D4 0880 642 CLRL -(SP)
OC AD D4 0882 643 CLRL 12(FP) ; Make this final frame in case we need it
1444'DF 04 00 OC 0885 644 PROBER #0,#4,@MISC+IMGDMP$R1
05 13 0888 645 BEQL 25$ ; The FP is no good
5D 1444'CF DD 088D 646 MOVL MISC+IMGDMP$R1,FP ; Real FP
5D DD 0892 647 25$: PUSHL FP
5D DD 0894 648 MOVL FP,R0
OC A0 04 00 OC 0897 649 30$: PROBER #0,#4,12(R0) ; In case of a corrupted stack
0B 13 089C 650 BEQL 40$
50 OC A0 DD 089E 651 MOVL 12(R0),R0 ; Trace back FP
05 13 08A2 652 BEQL 40$ ; End of the chain
6E 5D DD 08A4 653 MOVL R0,(SP) ; This is a good one
EE 11 08A7 654 BRB 30$
04 DD 08A9 655 40$: PUSHL #4
5E DD 08AB 656 PUSHL SP ; Mechanism array
51 DD 08AD 657 PUSHL R1 ; Signal array
02 DD 08AF 658 PUSHL #2
5C 5E DD 08B1 659 MOVL SP,AP ; Phony SS$_DEBUG frame
08B9'CF 6C FA 08B4 660 CALLG (AP),50$ ; DEBUG likes to see the resulting frame
OFAFC 08B9 661 50$: .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
OAAAC'CF 5C DD 08BB 662 MOVL AP,DBG_ARG+4
5C OAA8'CF 9E 08C0 663 MOVAB DBG_ARG,AP
50 0000'CF DD 08C5 664 MOVL SS$_NORMAL,R0 ; DEBUG is used to this
OF20'DF 17 08CA 665 JMP @DEBUG_BEG
08CE 666 ;
08CE 667 ;
08CE 668 ;
08CE 669 ; Get a new map block
08CE 670 ;
08CE 671 NEXT_MAP:
1394'CF 1398'CF DD 08CE 672 PUSHL DMP_RAB+RAB$R_UBF ; Save current VA
0200 8F B0 08D2 673 MOVW #512,DMP_RAB+RAB$R_USZ ; One block
1398'CF OF24'CF 9E 08D9 674 MOVAB MAP,DMP_RAB+RAB$R_UBF
08E0 675 $READ RAB = DMP_RAB
12 50 E9 08EB 676 BLBC R0,10$
13AC'CF D6 08EE 677 INCL DMP_RAB+RAB$R_BKT ; Next block
1398'CF 8E DD 08F2 678 MOVL (SPT+,DMP_RAB+RAB$R_UBF) ; Restore VA
55 OF24'CF 9E 08F7 679 MOVAB MAP,R5
52 85 DD 08FC 680 MOVL (R5)+,R2 ; Get next count
05 08FF 681 RSB
0900 682 ;
04 0900 683 10$: RET ; Error - give up
```



```
0901 685 :
0901 686 : Misc privileged routines
0901 687 :
0901 688 :
0901 689 :
0901 690 : Delete all of P0
0901 691 :
0901 692 DELETE:
0901 693 .WORD ^M<R2,R6>
0903 694 JSB @#EXE$RESETVEC ; Reset privileged library stuff
0909 695 $DELTVA_S INADR = ALL_P0
0918 696 :
0918 697 : Deallocate all image control blocks that describe currently activated images
0918 698 :
0918 699 MOVAQ G^IAC$GL_ICBFL,R2 ; R2 = address of free list
091F 700
091F 701 MOVAQ G^IAC$GL_IMAGE_LIST,R1 ; R1 = listhead of ICBS
51 00000000'GF 7E 091F 701 5$: REMQUE @<R1>,R6 ; Remove next entry
56 00 06 0F 0926 702 : List empty - all done
04 B2 66 0E 092A 703 BVS 10$ ; Insert at end of free list
F4 11 092C 704 INSQUE (R6),@4(R2) ; Go back for more
0930 705 BRB 5$
0932 706
0932 707 10$: RET
0933 708 :
0933 709 : Fix up the stack limit
0933 710 :
0933 711 FIX_STACK:
0933 712 .WORD 0
0000000C'9F 0A98'CF 0000 0935 713 MOVL STACK_INI,@#CTL$AL_STACK+12
04 093E 714 RET
093F 715 :
093F 716 : Fix up CTL$GL_IMGHDRBF
093F 717 :
093F 718 FIX_IMGHDRBF:
093F 719 .WORD 0
00000000'9F 0A9C'CF 0000 0941 720 MOVL IMGHDRBF_INI,@#CTL$GL_IMGHDRBF
04 094A 721 RET
094B 722 :
094B 723 : Reset privileges to get rid of image privileges
094B 724 :
094B 725 RESET_PRIV:
094B 726 .WORD 0
50 00000000'9F 0000 094B 726
00E8 C0 7C 094D 727 MOVL @#CTL$GL_PHD,R0
7E 00000004'9F D2 0954 728 CLRQ PHD$Q IMAGPRIV(R0) ; No more authorized image provs
7E 00000000'9F D2 0958 729 MCOML @#CTL$GL_PROCPRIV+4,-(SP) ; Complement of the permanent privileges
51 5E D0 095F 730 MCOML @#CTL$GL_PROCPRIV,-(SP)
0966 731 MOVL SP,R1
0969 732 $SETPRV_S ENBFLG = #0,-
0969 733 PRVADR = (R1)
04 0978 734 RET
0979 735 :
0979 736 : Reset privileged library vectors
0979 737 :
0979 738 RESET_VEC:
0979 739 .WORD ^M<R2,R6>
50 00000000'9F 0044 0979 739
00000000'8F D0 097B 740 JSB @#EXE$RESETVEC ; Reset privileged library stuff
0981 741 MOVL #SS$_NORMAL,R0
```



```
04 0988 742 RET
    0989 743 :
    0989 744 : IMGACT the original image and then reset the image privileges
    0989 745 :
    0989 746 IMGACT:
0000 0989 747 .WORD 0
    098B 748 $IMGACT_S NAME = IMAGE,- ; Activate original image
    098B 749 DFLNAM = IMG_DEFAULT,-
    098B 750 RETADR = IMG_RETADR,-
    098B 751 HDRBUF = IMG_HDRBUF
7E 50 DD 09A8 752 PUSHL R0
51 03 7D 09AA 753 MOVQ #<1@PRV$V_CMKRN!>,<1@PRV$V_CMEXEC>,-(SP) ; Restore privileges
    5E D0 09AD 754 MOVL SP,R1
    09B0 755 $SETPRV_S ENBFLG = #1,-
    09B0 756 PRVADR = (R1)
50 8E 7D 09BF 757 MOVQ (SP)+,R0 ; Clean up the stack
    50 8E D0 09C2 758 POPL R0
    04 09C5 759 RET
```

```
09C6 761 :
09C6 762 : Routines to handle misc address space
09C6 763 :
09C6 764 :
09C6 765 : A table is used
09C6 766 : count of table entries
09C6 767 : longword offset of size in MISC
09C6 768 : longword offset of file block in MISC
09C6 769 : longword offset of VA in MISC_VA
09C6 770 :
09C6 771 : Build table of VA's of misc pieces of address space
09C6 772 :
09C6 773 BLD_MISC VA:
50 1454'CF D0 09C6 774 MOVL MISC+IMGDMP$FREE_P0,R0 ; Start here
51 177F'CF 9E 09CB 775 MOVAB MISC_CONTROL,R1 ; Table address
52 81 D0 09D0 776 MOVL (R1)+,R2 ; Number of entries
53 53 61 D0 09D3 777 10$: MOVL (R1),R3 ; Offset for size
53 140C'CF43 D0 09D6 778 MOVL MISC[R3],R3 ; Size
54 11 13 09DC 779 BEQL 20$ ; Nothing saved for this one
54 08 A1 D0 09DE 780 MOVL 8(R1),R4 ; Offset for VA
171F'CF44 50 D0 09E2 781 MOVL R0,MISC_VA[R4] ; Save VA
53 53 09 78 09E8 782 ASHL #9,R3,R3 ; Page count -> bytes
50 53 C0 09EC 783 ADDL R3,R0 ; New VA
51 0C C0 09EF 784 20$: ADDL #12,R1 ; Next entry
DE 52 F5 09F2 785 SOBGTR R2,10$
171B'CF 50 D0 09F5 786 MOVL R0,DEBUG_VA ; Start DEBUG at the end
05 09FA 787 RSB
09FB 788 :
09FB 789 : Restore misc VA
09FB 790 :
09FB 791 RESTORE_MISC VA:
56 177F'CF 9E 09FB 792 MOVAB MISC_CONTROL,R6 ; Table address
57 171F'CF 9E 0A00 793 MOVAB MISC_VA,R7
58 140C'CF 9E 0A05 794 MOVAB MISC,R8
55 86 D0 0A0A 795 MOVL (R6)+,R5 ; Number of entries
52 66 D0 0A0D 796 10$: MOVL (R6),R2 ; Offset of area size
52 6842 D0 0A10 797 MOVL (R8)[R2],R2 ; Size
53 12 13 0A14 798 BEQL 20$ ; Nothing there
53 08 A6 D0 0A16 799 MOVL 8(R6),R3 ; Offset of VA
53 6743 D0 0A1A 800 MOVL (R7)[R3],R3 ; VA
54 04 A6 D0 0A1E 801 MOVL 4(R6),R4 ; Offset of file block
54 6844 D0 0A22 802 MOVL (R8)[R4],R4 ; File block
07 10 0A26 803 BSBB READ_ONE_VA
56 0C C0 0A28 804 20$: ADDL #12,R6 ; Next entry
DF 55 F5 0A2B 805 SOBGTR R5,10$
05 0A2E 806 RSB
0A2F 807 :
0A2F 808 :
0A2F 809 : Read a piece of address space
0A2F 810 : R2 = page count
0A2F 811 : R3 = starting VA
0A2F 812 : R4 = starting file block
0A2F 813 :
0A2F 814 READ_ONE VA:
13AC'CF 54 D0 0A2F 815 MOVL R4,DMP_RAB+RAB$BKT
1398'CF 53 D0 0A34 816 10$: MOVL R3,DMP_RAB+RAB$BKT-UBF ; Address
160C'CF 53 D0 0A39 817 MOVL R3,CREATE_PAGE ; Create the address range
```



```
0000FE00 54 52 D0 0A3E 818
          8F 54 D1 0A41 819
          07 54 1B 0A48 820
54 0000007F 8F D0 0A4A 821
   54 54 09 78 0A51 822 20$:
 1394'CF 54 B0 0A55 823
          54 D7 0A5A 824
1610'CF 53 54 C1 0A5C 825
          23 50 E9 0A71 827
          15 50 E9 0A7F 829
          54 D6 0A82 830
          53 54 C0 0A84 831
54 54 F7 8F 78 0A87 832
 13AC'CF 54 C0 0A8C 833
          52 54 C2 0A91 834
          9E 14 0A94 835
          05 0A96 836
          04 0A97 837 50$:
```

```
MOVL R2,R4 ; Remaining page count
CMPL R4,#<127*512> ; Compare with maximum
BLEQU 20$
MOVL #127,R4 ; Use maximum
ASHL #9,R4,R4
MOVW R4,DMP_RAB+RAB$W_USZ ; Byte count
DECL R4
ADDL3 R4,R3,CREATE_PAGE+4
$CRETVA_S INADR = CREATE_PAGE
BLBC R0,50$
$READ RAB = DMP_RAB
BLBC R0,50$
INCL R4
ADDL R4,R3 ; Update address
ASHL #-9,R4,R4 ; Get back the page count
ADDL R4,DMP_RAB+RAB$W_USZ
SUBL R4,R2 ; Any pages left?
BGTR 10$ ; Continue
RSB ; Error
RET
```



```
0A98 839 ;
0A98 840 ; Data
0A98 841 ;
0A98 842 ;
0A98 843 STACK_INI: ; Stack limit before $IMGACT
00000000 0A98 844 .LONG 0
0A9C 845 ;
0A9C 846 IMGHDRBF_INI: ; CTL$GL_IMGHDRBF before our IMGACTs
00000000 0A9C 847 .LONG 0
0AA0 848 ;
3FFFFFFF 00000000 0AA0 849 ALL_PO: .LONG 0,*X3FFFFFFF ; Range to delete all of PO
0AA8 850 ;
00000000 00000000 00000000 00000006 0AA8 851 DBG_ARG: ; Argument list to call DEBUG
00000000 00000000 00000000 00000000 0AA8 852 .LONG 6,0,0,0,0,0,CLIM_DBGEXCP
0AC4 853 ;
59 53 24 53 59 53 00000ACC'010E0000' 0AC4 854 IMG_DEFAULT: ; Default name for images
45 58 45 2E 3A 4D 45 54 53 0AC4 855 .ASCID /SYS$SYSTEM:.EXE/
0ADB 856 ;
49 4C 24 53 59 53 00000AE3'010E0000' 0ADB 857 DBG_DEFAULT: ; Default name for DEBUG
45 58 45 2E 3A 59 52 41 52 42 0ADB 858 .ASCID /SYS$LIBRARY:.EXE/
0AE9 859 ;
00000000 00000000 0AF3 860 IMAGE: .LONG 0,0 ; Descriptor for image name
0AFB 861 ;
00000B03 0AFB 862 IMG_RETADR: ; Address range of image
0B03 863 .BLKL 2
00000D03 0B03 864 ;
0B03 865 IMG_HDRBUF: ; IMGACT buffer for image
0D03 866 .BLKB 512
3FFFFFFF 00000000 0D03 867 ;
0D03 868 DBG_RANGE: ; Range to merge in DEBUG
0D0B 869 .LONG 0,*X3FFFFFFF
47 55 42 45 44 00000D13'010E0000' 0D0B 870 ;
0D18 871 DEBUG: .ASCID /DEBUG/ ; Name of DEBUG
0D18 872 ;
00000D20 0D18 873 DBG_RETADR: ; Address range used by DEBUG
0D20 874 .BLKL 2
00000F20 0D20 875 ;
0F20 876 DBG_HDRBUF: ; IMGACT buffer for DEBUG
0F20 877 .BLKB 512
00000F24 0F20 878 ;
0F24 879 DEBUG_BEG: ; Transfer address for DEBUG
00001124 0F24 880 .BLKL 1
0F24 881 ;
00001124 0F24 882 MAP: .BLKB 512 ; Address map buffer
1124 883 ;
00001324 1124 884 IMGHDR:: ; First block of image header
1124 885 .BLKB 512
1324 886 ;
1324 887 .ALIGN LONG
1324 888 ;
1324 889 DMP_FAB: $FAB
1324 890 FAC = <BIO,GET>,-
1324 891 FNA = NAME_BUFFER,-
1324 892 DNA = DFLNAM,-
1324 893 DNS = 4
```



```

1374 893 ;
1374 894 DMP_RAB: $RAB FAB = DMP_FAB,-
1374 895 USZ = 512,-
1374 896 ROP = BIO
50 4D 44 2E 13B8 897 DFLNAM: .ASCII /.DMP/ ; Default name for dumps
13BC 898 ;
13BC 899 DUMP_NAME: ; Descriptor for name of dump file
00000040 13BC 900 .LONG 64
000013C4 13C0 901 .ADDRESS NAME_BUFFER
13C4 902 NAME_BUFFER:
00001404 13C4 903 .BLKB 64
1404 904 ;
0000140C 1404 905 NEW_PO: .BLKL 2 ; P0 space for relocating code
140C 906 ;
0000160C 140C 907 MISC:: .BLKB 512 ; Buffer for misc data block
160C 908 ;
00001614 160C 909 CREATE_PAGE: ; Address range for a page to be created
1614 910 .BLKL 2
1614 911 ;
000000FF 1614 912 IMAGE_DESC: ; Descriptor for image name
0000161C 1618 913 .LONG NAM$C_MAXRSS
0000171B 161C 914 .ADDRESS 1$
171B 915 1$: .BLKB NAM$C_MAXRSS
171B 916 ;
00000000 171B 917 DEBUG_VA: ; VA for DEBUG
171F 918 .LONG 0
171F 919 ;
0000177F 171F 920 MISC_VA:: ; VA's of misc data areas
177F 921 .BLKL IMGMOV$C_LENGTH
177F 922 ;
00000006 177F 923 MISC_CONTROL: ; Description of misc address space
0000002E 1783 924 .LONG <1$ - MISC_CONTROL>/12
0000002C 1787 925 .LONG IMGDMP$C_KSTK_SIZ/4 ; Kernel stack
00000000 178B 926 .LONG IMGDMP$C_KSTK_BLK/4
00000031 178F 927 .LONG IMGMOV$C_KSTK74
0000002F 1793 928 .LONG IMGDMP$C_ESTK_SIZ/4 ; Exec stack
00000001 1797 929 .LONG IMGDMP$C_ESTK_BLK/4
00000034 179B 930 .LONG IMGMOV$C_ESTK74
00000032 179F 931 .LONG IMGDMP$C_VECPAG_SIZ/4 ; Vector page
00000002 17A3 932 .LONG IMGDMP$C_VECPAG_BLK/4
00000037 17A7 933 .LONG IMGMOV$C_VECPAG74
00000035 17AB 934 .LONG IMGDMP$C_PIO_SIZ/4 ; PIO (RMS) area
00000003 17AF 935 .LONG IMGDMP$C_PIO_BLK/4
0000003A 17B3 936 .LONG IMGMOV$C_PIO74
00000038 17B7 937 .LONG IMGDMP$C_IMGCTX_SIZ/4 ; Image activator context pages
00000004 17BB 938 .LONG IMGDMP$C_IMGCTX_BLK/4
0000003D 17BF 939 .LONG IMGMOV$C_IMGCTX74
0000003B 17C3 940 .LONG IMGDMP$C_USRCTX_SIZ/4 ; User writeable context pages
00000005 17C7 941 .LONG IMGDMP$C_USRCTX_BLK/4
17CB 942 .LONG IMGMOV$C_USRCTX74
17CB 943 1$:
17CB 944 ;
000017D3 17CB 945 THIS_HDR: ; Pointers to this image's header buffer
17D3 946 .BLKL 2
17D3 947 ;
000018D2 17D3 948 IMGNAME$S: ; ESS from open of image
17D3 949 .BLKB NAM$C_MAXRSS
```



```
18D2 950 ;
00000000' 18D2 951 CTRY_DISABL: ; Mask to disable control-Y
18D2 952 .LONG LIBSM_CLI_CTRLY
18D6 953 ;
00000000' 18D6 954 OLD_CTRL: ; Control-Y state
18D6 955 .LONG 0
18DA 956 ;
00000000' 18DA 957 INP_CHAN: ; SYSS$INPUT channel
18DA 958 .LONG 0
18DE 959 ;
00000000' 18DE 960 SUBP_PID: ; PID of subprocess
18DE 961 .LONG 0
18E2 962 ;
18E2 963 MOVE_END: ; End of code to be relocated
18E2 964 ;
18E2 965 ;
18E2 966 ; Data after this point is not relocated
18E2 967 ;
18E2 968 ;
31 50 000018EA'010E0000' 18E2 969 CLI_PARAMETER: ; Get the command line parameter
18E2 970 .ASCID /P1/
18EC 971 ;
45 47 41 4D 49 000018F4'010E0000' 18EC 972 CLI_IMAGE: ; The /IMAGE qualifier
18EC 973 .ASCID /IMAGE/
18F9 974 ;
47 41 4D 49 4F 4E 00001901'010E0000' 18F9 975 CLI_NOIMAGE: ; The /NOIMAGE qualifier
18F9 976 .ASCID /NOIMAGE/
45 1907
1908 977 ;
4E 49 24 53 59 53 00001910'010E0000' 1908 978 SYSS$INPUT: ; Stings for TRNLOG
54 55 50 1908 979 .ASCID /SYSS$INPUT/
1916
55 4F 24 53 59 53 00001921'010E0000' 1919 980 SYSS$OUTPUT:
54 55 50 54 1919 981 .ASCID /SYSS$OUTPUT/
1927
52 45 24 53 59 53 00001933'010E0000' 192B 982 SYSS$ERROR:
52 4F 52 192B 983 .ASCID /SYSS$ERROR/
1939
193C 984 ;
00000040 193C 985 INPUT_TRN: ; Translation of initial SYSS$INPUT
00001944' 1940 986 .LONG 64
1944 987 .ADDRESS SYSS$INPUT_TRN
00001984 1944 988 SYSS$INPUT_TRN:
1984 989 .BLKB 64
1984 990 ;
00000040 1984 991 INPUT: .LONG 64 ; Output strings from TRNLOG
0000198C' 1988 992 .ADDRESS 1$
000019CC 198C 993 1$: .BLKB 64
00000040 19CC 994 OUTPUT: .LONG 64
000019D4' 19D0 995 .ADDRESS 1$
00001A14 19D4 996 1$: .BLKB 64
00000040 1A14 997 ERROR: .LONG 64
00001A1C' 1A18 998 .ADDRESS 1$
00001A5C 1A1C 999 1$: .BLKB 64
1A5C 1000 ;
00000000 1A5C 1001 INP_MBX: ; Channel for communications mailbox
1A5C 1002 .LONG 0
```



```
00000000 1A60 1003 ;  
1A60 1004 TERM_MBX: ; Channel for termination mailbox  
1A60 1005 .LONG 0  
00000000 1A64 1006 ;  
1A64 1007 INP_MBX_UNIT: ; Unit number for communications mailbox  
1A64 1008 .LONG 0  
1A68 1009 ;  
00000000 1A68 1010 TERM_MBX_UNIT: ; Unit number for termination mailbox  
1A68 1011 .LONG 0  
1A6C 1012 ;  
41 42 4D 5F 1B 1B 00001A74'010E0000' 1A6C 1013 INPFA0: .ASCID <27><27>/_MBA!5ZW:/ ; FA0 string for mailbox name + ESCs  
3A 57 5A 35 21 1A7A  
1A7F 1014 ;  
0000000E 1A7F 1015 INP_MBX_NAM: ; Communications mailbox name  
1A7F 1016 .LONG 14  
00001A87' 1A83 1017 .ADDRESS 1$  
00001A97 1A87 1018 1$: .BLKB 16  
1A97 1019 ;  
59 53 24 53 59 53 00001A9F'010E0000' 1A97 1020 ANAL_IMG: ; Name of this image for CREPRC  
44 4D 49 4C 41 4E 41 3A 4D 45 54 53 1A97 1021 .ASCID /SYSS$SYSTEM:ANALIMDMP.EXE/  
45 58 45 2E 50 4D 1AA5  
1AB1  
1AB7 1022 ;  
00001AC1 1AB7 1023 TERM_MSG: ; Termination mailbox message  
1AB7 1024 .BLKB 10  
1AC1 1025 ;  
00001AC9 1AC1 1026 LOG_OUT: ; Descriptor for output logical name  
1AC1 1027 .BLKL 2  
1AC9 1028 ;  
00001AD1 1AC9 1029 LOG_IN: ; Descriptor for input logical name  
1AC9 1030 .BLKL 2  
1AD1 1031 ;  
00000074 1AD1 1032 MBXCHAR: ;  
00001AD9' 1AD1 1033 .LONG DIB$K_LENGTH  
1AD5 1034 .ADDRESS MBXCHARBUF  
1AD9 1035 MBXCHARBUF: ; Buffer for mailbox characteristics  
00001B4D 1AD9 1036 .BLKB DIB$K_LENGTH  
1B4D 1037 ;  
00001B55 1B4D 1038 IOSB: .BLKL 2 ; IOSB for mailbox use  
1B55 1039 ;  
00000040 1B55 1040 REAL_INPUT: ; Real SYSS$INPUT when in subprocess  
00001B5D' 1B55 1041 .LONG 64  
00001B9D 1B59 1042 .ADDRESS 1$  
1B5D 1043 1$: .BLKB 64  
1B9D 1044 ;  
1B9D 1045 .ALIGN LONG  
1BA0 1046 ;  
1BA0 1047 OUTFAB: $FAB FNM = <SYSS$OUTPUT>,- ; FAB for SYSS$OUTPUT  
1BA0 1048 FAC = <GET,PUT>,-  
1BA0 1049 FOP = <CIF>  
1BF0 1050 ;  
1BF0 1051 OUTRAB: $RAB FAB = OUTFAB  
1C34 1052 ;  
1C34 1053 IMGFAB: $FAB DNM = <.EXE>,-  
1C34 1054 NAM = IMGNAM  
1C84 1055 ;  
1C84 1056 IMGNAM: $NAM ESS = NAM$C_MAXRSS,-
```


ANALIMDMP
V04-000

E 9

16-SEP-1984 01:41:09 VAX/VMS Macro V04-00 Page 24
5-SEP-1984 01:28:48 [IMGDMP.SRC]ANALIMDMP.MAR;1 (11)

```

72 43 0A 0D 0A 0D 00001CEC'010E0000' 1C84 1057
62 75 73 20 61 20 67 6E 69 74 61 65 1CE4 1058 :
73 73 65 63 6F 72 70 1CE4 1059 CRE_SUB_MSG: .ASCID <13><10><13><10>/Creating a subprocess/
1CF2
1CFE
1D05 1060 :
1D05 1061 COND_MSG: .ASCID <13><10>/Condition signalled to take dump:/
1D13
1D1F
1D2B
1D30 1062 :
1D30 1063 .END ANALIMDMP
```


ANALIMDMP
Symbol table

F 9

16-SEP-1984 01:41:09 VAX/VMS Macro V04-00 Page 25
5-SEP-1984 01:28:48 [IMGDMP.SRC]ANALIMDMP.MAR;1 (11)

\$\$TAB	= 00001C84	R	01
\$\$TABEND	= 00001CE4	R	01
\$\$TMP	= 00000000		
\$\$TMP1	= 00000001		
\$\$TMP2	= 000000CF		
\$\$TMPX	= 0000000A	R	03
\$\$TMPX1	= 00000004		
\$\$T1	= 00000001		
\$\$T2	= 00000004		
ALL_PO	00000AA0	R	01
ANALIMDMP	00000000	R	01
ANAL_IMG	00001A97	R	01
BLD_MISC_VA	000009C6	R	01
CLISGET_VALUE	*****	X	01
CLISM_DBGEXCP	*****	X	01
CLISPRESENT	*****	X	01
CLI_IMAGE	000018EC	R	01
CLI_NOIMAGE	000018F9	R	01
CLI_PARAMETER	000018E2	R	01
CNTRL_Y_AST	000003F5	R	01
COND_MSG	00001D05	R	01
CREATE_PAGE	0000160C	R	01
CREATE_SUBP	0000021A	R	01
CRE_SUB_MSG	00001CE4	R	01
CTLSAL_STACK	*****	X	01
CTLSGL_IMGHDRBF	*****	X	01
CTLSGL_PHD	*****	X	01
CTLSGL_PROCPRIV	*****	X	01
CTRY_DISABL	000018D2	R	01
DBG_ARG	00000AA8	R	01
DBG_DEFAULT	00000ADB	R	01
DBG_HDRBUF	00000D20	R	01
DBG_RANGE	00000D03	R	01
DBG_RETADR	00000D18	R	01
DEBUG	00000D0B	R	01
DEBUG_BEG	00000F20	R	01
DEBUG_VA	0000171B	R	01
DELETE	00000901	R	01
DFLNAM	000013B8	R	01
DIBSK_LENGTH	= 00000074		
DIBSW_UNIT	= 0000000C		
DISPLAY_DUMP	*****	X	01
DMP_FAB	00001324	R	01
DMP_RAB	00001374	R	01
DUMP_NAME	000013BC	R	01
ERROR	00001A14	R	01
EXESRESETVEC	*****	X	01
FABSB_DNS	= 00000035		
FABSB_FNS	= 00000034		
FABSC_BID	= 00000003		
FABSC_BLN	= 00000050		
FABSC_SEQ	= 00000000		
FABSC_VAR	= 00000002		
FABSL_ALQ	= 00000010		
FABSL_DNA	= 00000030		
FABSL_FNA	= 0000002C		
FABSL_FOP	= 00000004		

FABSV_BIO	= 00000005		
FABSV_CHAN_MODE	= 00000002		
FABSV_CIF	= 00000019		
FABSV_FILE_MODE	= 00000004		
FABSV_GET	= 00000001		
FABSV_LNM_MODE	= 00000000		
FABSV_PUT	= 00000000		
FABSW_GBC	= 00000048		
FIX_IMGHDRBF	0000093F	R	01
FIX_STACK	00000933	R	01
GET_DEBUG	000007BB	R	01
GET_P1	000006DB	R	01
GET_TRAN	000004FB	R	01
IAC\$GL_ICBFL	*****	X	01
IAC\$GL_IMAGE_LIST	*****	X	01
IAC\$M_MERGE	= 00000010		
IMDSW_IMGIDOFF	= 00000006		
IHIST_IMGNAM	= 00000000		
IMAGE	00000AF3	R	01
IMAGE_DESC	00001614	R	01
IMGACT	00000989	R	01
IMGDMP	= 00000000		
IMGDMP\$C_LENGTH	= 000000F8		
IMGDMP\$C_VERSION	= 00000003		
IMGDMP\$AL	= 00000034		
IMGDMP\$AL_ASTACK	= 00000050		
IMGDMP\$AL_ASTACKCNT	= 000000A0		
IMGDMP\$AL_ASTACKEN	= 00000054		
IMGDMP\$AL_ASTACKLM	= 000000A4		
IMGDMP\$AL_BIOCNT	= 00000068		
IMGDMP\$AL_BIOLM	= 0000006C		
IMGDMP\$AL_BUFIO	= 00000070		
IMGDMP\$AL_BYTCNT	= 00000074		
IMGDMP\$AL_BYTLM	= 00000078		
IMGDMP\$AL_DIOCNT	= 0000007C		
IMGDMP\$AL_DIOLM	= 00000080		
IMGDMP\$AL_DIRIO	= 00000084		
IMGDMP\$AL_EFCS	= 00000060		
IMGDMP\$AL_EFCU	= 00000064		
IMGDMP\$AL_ENQCN	= 000000A8		
IMGDMP\$AL_ENQLM	= 000000AC		
IMGDMP\$AL_ESTK_BLK	= 000000BC		
IMGDMP\$AL_ESTK_SIZ	= 000000C4		
IMGDMP\$AL_ESTK_VA	= 000000C0		
IMGDMP\$AL_FILCNT	= 00000088		
IMGDMP\$AL_FILLM	= 0000008C		
IMGDMP\$AL_FIRST_MAP	= 0000003C		
IMGDMP\$AL_FP	= 00000038		
IMGDMP\$AL_FREE_PO	= 00000048		
IMGDMP\$AL_FREE_P1	= 0000004C		
IMGDMP\$AL_IMGCTX_BLK	= 000000E0		
IMGDMP\$AL_IMGCTX_SIZ	= 000000E8		
IMGDMP\$AL_IMGCTX_VA	= 000000E4		
IMGDMP\$AL_KSTK_BLK	= 000000B0		
IMGDMP\$AL_KSTK_SIZ	= 000000B8		
IMGDMP\$AL_KSTK_VA	= 000000B4		
IMGDMP\$AL_PIO_BLK	= 000000D4		

ANALIMDMP
Symbol table

G 9

16-SEP-1984 01:41:09 VAX/VMS Macro V04-00 Page 26
5-SEP-1984 01:28:48 [IMGDMP.SRC]ANALIMDMP.MAR;1 (11)

IMGDMP\$P_IO_SIZ	= 000000DC		
IMGDMP\$P_IO_VA	= 000000D8		
IMGDMP\$P_PRCNT	= 00000090		
IMGDMP\$P_PRCM	= 00000094		
IMGDMP\$P_R0	= 00000000		
IMGDMP\$P_R1	= 00000004		
IMGDMP\$P_R10	= 00000028		
IMGDMP\$P_R11	= 0000002C		
IMGDMP\$P_R2	= 00000008		
IMGDMP\$P_R3	= 0000000C		
IMGDMP\$P_R4	= 00000010		
IMGDMP\$P_R5	= 00000014		
IMGDMP\$P_R6	= 00000018		
IMGDMP\$P_R7	= 0000001C		
IMGDMP\$P_R8	= 00000020		
IMGDMP\$P_R9	= 00000024		
IMGDMP\$P_SP	= 00000030		
IMGDMP\$P_TQCNT	= 00000098		
IMGDMP\$P_TQLM	= 0000009C		
IMGDMP\$P_USRCTX_BLK	= 000000EC		
IMGDMP\$P_USRCTX_SIZ	= 000000F4		
IMGDMP\$P_USRCTX_VA	= 000000F0		
IMGDMP\$P_USRSTK	= 00000040		
IMGDMP\$P_VECPAG_BLK	= 000000C8		
IMGDMP\$P_VECPAG_SIZ	= 000000D0		
IMGDMP\$P_VECPAG_VA	= 000000CC		
IMGDMP\$P_VERSION	= 00000044		
IMGDMP\$Q_CURPRIV	= 00000058		
IMGDMP\$S_CURPRIV	= 00000008		
IMGDMP\$S_IMGDMF	= 000000F8		
IMGFAB	00001C34	R	01
IMGHDR	00001124	RG	01
IMGHDRBF_INI	00000A9C	R	01
IMGMOV	= 00000000		
IMGMOV\$C_LENGTH	= 00000018		
IMGMOV\$C_ESTK	= 00000004		
IMGMOV\$C_IMGCTX	= 00000010		
IMGMOV\$C_KSTK	= 00000000		
IMGMOV\$C_PIO	= 0000000C		
IMGMOV\$C_USRCTX	= 00000014		
IMGMOV\$C_VECPAG	= 00000008		
IMGMOV\$S_IMGMOV	= 00000018		
IMGNAM	00001C84	R	01
IMGNAME\$S	000017D3	R	01
IMG_DEFAULT	00000AC4	R	01
IMG_HDRBUF	00000B03	R	01
IMG_RETADR	00000AFB	R	01
INIT SUBP	0000041F	R	01
INPF\$O	00001A6C	R	01
INPUT	00001984	R	01
INPUT TRN	0000193C	R	01
INP_CHAN	000018DA	R	01
INP_MBX	00001A5C	R	01
INP_MBX_NAM	00001A7F	R	01
INP_MBX_UNIT	00001A64	R	01
IOSM_CTRLVAST	= 00000080		
IOSM_NOW	= 00000040		

IOS_READVBLK	= 00000031		
IOS_SETMODE	= 00000023		
IOS_WRITEVBLK	= 00000030		
IOSB	00001B4D	R	01
LIB\$DISABLE_CTRL	*****	X	01
LIB\$ENABLE_CTRL	*****	X	01
LIB\$M_CLI_CTRLV	*****	X	01
LOG_IN	00001AC9	R	01
LOG_OUT	00001AC1	R	01
MAP	00000F24	R	01
MBXCHAR	00001AD1	R	01
MBXCHARBUF	00001AD9	R	01
MBX_UNIT	00000542	R	01
MISC	0000140C	RG	01
MISC_CONTROL	0000177F	R	01
MISC_VA	0000171F	RG	01
MOVE_BEG	00000561	R	01
MOVE_END	000018E2	R	01
NAM\$B_ESL	= 0000000B		
NAM\$B_ESS	= 0000000A		
NAM\$B_NOP	= 00000008		
NAM\$B_RSS	= 00000002		
NAM\$C_BID	= 00000002		
NAM\$C_BLN	= 00000060		
NAM\$C_MAXRSS	= 000000FF		
NAM\$C_ESA	= 0000000C		
NAM\$C_RSA	= 00000004		
NAME_BUFFER	000013C4	R	01
NEW_PO	00001404	R	01
NEXT_MAP	000008CE	R	01
OLD_CTRL	000018D6	R	01
OUTFAB	00001BA0	R	01
OUTPUT	000019CC	R	01
OUTRAB	00001BF0	R	01
PHD\$Q_IMAGPRIV	= 000000E8		
PRV\$V_CMEXEC	= 00000001		
PRV\$V_CMKRN	= 00000000		
RAB\$B_RAC	= 0000001E		
RAB\$C_BID	= 00000001		
RAB\$C_BLN	= 00000044		
RAB\$C_SEQ	= 00000000		
RAB\$C_CTX	= 00000018		
RAB\$C_ROP	= 00000004		
RAB\$V_BIO	= 0000000B		
READ_ONE_VA	00000A2F	R	01
REAL_INPOT	00001B55	R	01
RESET_PRIV	0000094B	R	01
RESET_VEC	00000979	R	01
RESTORE_MISC_VA	000009FB	R	01
RM\$S_FNF	*****	X	01
SS\$_BADFILEVER	*****	X	01
SS\$_DEBUG	*****	X	01
SS\$_NORMAL	*****	X	01
SS\$_NOTRAN	*****	X	01
SS\$_VAFULL	*****	X	01
STACK_INI	00000A98	R	01
SUBP_PID	000018DE	R	01

ANALIMDMP
Symbol table

SYSS\$ASSIGN	*****	GX	01
SYSS\$CLOSE	*****	GX	01
SYSS\$CMEXEC	*****	GX	01
SYSS\$CMKRN	*****	GX	01
SYSS\$CONNECT	*****	GX	01
SYSS\$CREATE	*****	GX	01
SYSS\$CRELOG	*****	GX	01
SYSS\$CREMBX	*****	GX	01
SYSS\$CREPRC	*****	GX	01
SYSS\$CRETVA	*****	GX	01
SYSS\$DELPRC	*****	GX	01
SYSS\$DELTVA	*****	GX	01
SYSS\$DISCONNECT	*****	GX	01
SYSS\$ERROR	0000192B	R	01
SYSS\$EXIT	*****	GX	01
SYSS\$EXPREG	*****	GX	01
SYSS\$FAO	*****	X	01
SYSS\$GETCHN	*****	GX	01
SYSS\$IMGACT	*****	GX	01
SYSS\$IMGFIX	*****	GX	01
SYSS\$INPUT	00001908	R	01
SYSS\$INPUT_TRN	00001944	R	01
SYSS\$OPEN	*****	GX	01
SYSS\$OUTPUT	00001919	R	01
SYSS\$PUT	*****	GX	01
SYSS\$PUTMSG	*****	GX	01
SYSS\$QIO	*****	GX	01
SYSS\$QIOW	*****	GX	01
SYSS\$READ	*****	GX	01
SYSS\$SETPRV	*****	GX	01
SYSS\$TRNLOG	*****	GX	01
TERM_MBX	00001A60	R	01
TERM_MBX_UNIT	00001A68	R	01
TERM_MSG	00001AB7	R	01
THIS_HDR	000017CB	R	01

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes													
. ABS .	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE			
ANALIMDMP	00001D30 (7472.)	01 (1.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	LONG			
\$ABSS	00000000 (0.)	02 (2.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE			
\$RMSNAM	0000000E (14.)	03 (3.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE			

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	36	00:00:00.09	00:00:01.23
Command processing	139	00:00:00.77	00:00:03.81
Pass 1	488	00:00:19.26	00:00:42.53
Symbol table sort	0	00:00:02.09	00:00:04.22

Pass 2	198	00:00:04.37	00:00:09.38
Symbol table output	34	00:00:00.22	00:00:00.49
Psect synopsis output	5	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	903	00:00:26.84	00:01:01.69

The working set limit was 1500 pages.

105212 bytes (206 pages) of virtual memory were used to buffer the intermediate code.

There were 80 pages of symbol table space allocated to hold 1373 non-local and 71 local symbols.

1063 source lines were read in Pass 1, producing 29 object records in Pass 2.

65 pages of virtual memory were used to define 55 macros.

! Macro library statistics !

Macro library name	Macros defined
-----	-----
_\$255\$DUA28:[IMGDMP.OBJ]IMGDMPLIB.MLB;1	2
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	3
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	47
TOTALS (all libraries)	52

1781 GETS were required to define 52 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:ANALIMDMP/OBJ=OBJ\$:ANALIMDMP MSRC\$:ANALIMDMP/UPDATE=(ENH\$:ANALIMDMP)+EXECMLS/LIB+LIB\$:IMGDMPLIB/LIB

0186 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

